

# Package ‘rWCVP’

February 16, 2023

**Type** Package

**Title** Generating Summaries, Reports and Plots from the World Checklist of Vascular Plants

**Version** 1.2.4

**Maintainer** Matilda Brown <m.brown2@kew.org>

**Description** A companion to the World Checklist of Vascular Plants (WCVP). It includes functions to generate maps and species lists, as well as match names to the WCVP. For more details and to cite the package, see: Brown M.J.M., Walker B.E., Black N., Govaerts R., Ondo I., Turner R., Nic Lughadha E. (in press).  
“rWCVP: A companion R package to the World Checklist of Vascular Plants”. New Phytologist.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**LazyDataCompression** xz

**RoxygenNote** 7.2.3

**Imports** cli, dplyr, ggplot2, gt, htmltools, magrittr, phonics, purrr, RecordLinkage, rlang, rmarkdown, sf, stringr, tibble, tidyr, withr

**Suggests** rWCVPdata (>= 0.1.0), knitr, testthat (>= 3.0.0), spdep

**Additional\_repositories** <https://matildabrown.github.io/drat>

**VignetteBuilder** knitr

**URL** <https://github.com/matildabrown/rWCVP>,  
<https://matildabrown.github.io/rWCVP/>

**BugReports** <https://github.com/matildabrown/rWCVP/issues>

**Config/testthat/edition** 3

**Depends** R (>= 3.5.0)

**NeedsCompilation** no

**Author** Matilda Brown [aut, cre],  
Barnaby Walker [aut]

**Repository** CRAN

**Date/Publication** 2023-02-16 15:20:02 UTC

## R topics documented:

get_area_name . . . . .	2
get_wgsrpd3_codes . . . . .	3
powo_map . . . . .	4
powo_pal . . . . .	4
redlist_example . . . . .	5
taxonomic_mapping . . . . .	6
wcvp_checklist . . . . .	7
wcvp_distribution . . . . .	8
wcvp_distribution_map . . . . .	10
wcvp_match_exact . . . . .	11
wcvp_match_fuzzy . . . . .	12
wcvp_match_names . . . . .	13
wcvp_occ_mat . . . . .	15
wcvp_reformat . . . . .	17
wcvp_summary . . . . .	17
wcvp_summary_gt . . . . .	19
wgsrpd3 . . . . .	19
wgsrpd_mapping . . . . .	20

<b>Index</b>	<b>21</b>
--------------	-----------

---

get_area_name	<i>Get area description from vector of area codes</i>
---------------	---

---

### Description

Get area description from vector of area codes

### Usage

```
get_area_name(area_codes)
```

### Arguments

area\_codes      Character vector containing the set of codes to be mapped to a name.

### Details

Usually used as an inverse function for `get_wgsrpd3_codes`. Useful for condensing sets of codes for e.g. file names, plotting and table formatting.

**Value**

Character. Either a vector of length one, with a name for the set of Level 3 areas, or (if no name exists for that set of areas) the input vector of codes.

**Examples**

```
get_area_name(get_wgsrpd3_codes("Brazil"))
```

---

```
get_wgsrpd3_codes      Extract WGSRPD Level 3 (area) codes.
```

---

**Description**

Extract WGSRPD Level 3 (area) codes.

**Usage**

```
get_wgsrpd3_codes(geography, include_equatorial = NULL)
```

**Arguments**

geography	Character. The geography to convert into Level 3 codes. May be a WGSRPD area (Level 3), region (Level 2) or continent (Level 1), country (political) or hemisphere ("Northern Hemisphere", "Southern Hemisphere" or "Equatorial")
include_equatorial	Logical. Include Level 3 areas that span the equator? Defaults to NULL, which generates a message and includes these areas. Ignored if geography is not a hemisphere.

**Details**

Country mapping follows Gallagher et al. (2020). Importantly, this means that some overseas territories are not considered part of the country in this system, e.g. the Canary Islands are designated as their own Level 3 area, rather than part of Spain in this mapping. Where this is ambiguous, the mapping can be explored using `View(wgsrpd_mapping)`.

Gallagher, R. V., Allen, S., Rivers, M. C., Allen, A. P., Butt, N., Keith, D., & Adams, V. M. (2020). Global shortfalls in extinction risk assessments for endemic flora. *bioRxiv*, 2020.2003.2012.984559. <https://doi.org/10.1101/2020.03.12.984559>

**Value**

Character with area codes (Level 3) that fall within the geography.

**Examples**

```
get_wgsrpd3_codes("Brazil")
```

---

powo_map	<i>Plot a POWO style map for given range and range centroids.</i>
----------	---

---

**Description**

Plot a POWO style map for given range and range centroids.

**Usage**

```
powo_map(range_sf, centroids_sf)
```

**Arguments**

range_sf	A simple features (sf) data frame of range polygons
centroids_sf	A simple features (sf) data frame of range centroids

**Value**

A ggplot map of the range

---

powo_pal	<i>POWO colour palette for range maps</i>
----------	---

---

**Description**

Range maps displayed on the **POWO** website have a fixed, discrete colour palette based on the type of taxon occurrence in a region.

**Usage**

```
powo_pal()

scale_color_powo(...)

scale_colour_powo(...)

scale_fill_powo(...)
```

**Arguments**

...	Arguments passed on to <a href="#">discrete_scale</a>
palette	A palette function that when called with a single integer argument (the number of levels in the scale) returns the values that they should take (e.g., <a href="#">scales::hue_pal()</a> ).
limits	One of:

- NULL to use the default scale values
  - A character vector that defines possible values of the scale and their order
  - A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang [lambda](#) function notation.
- `drop` Should unused factor levels be omitted from the scale? The default, `TRUE`, uses the levels that appear in the data; `FALSE` uses all the levels in the factor.
- `na.translate` Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify `na.translate = FALSE`.
- `scale_name` The name of the scale that should be used for error messages associated with this scale.
- `name` The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If `NULL`, the legend title will be omitted.
- `labels` One of:
- `NULL` for no labels
  - `waiver()` for the default labels computed by the transformation object
  - A character vector giving labels (must be same length as breaks)
  - An expression vector (must be the same length as breaks). See `?plot-math` for details.
  - A function that takes the breaks as input and returns labels as output. Also accepts rlang [lambda](#) function notation.
- `guide` A function used to create a guide or its name. See [guides\(\)](#) for more information.
- `super` The super class to use for the constructed scale

**Value**

Character. Vector of names and HEX values to match those of **POWO**.

---

<code>redlist_example</code>	<i>Example dataset for name matching</i>
------------------------------	--

---

**Description**

A dataset containing 20 sampled Red List assessments for name-matching

**Usage**

```
redlist_example
```

**Format**

A data frame with 20 rows and 4 variables:

**assessmentId** Red List identifier

**scientificName** Taxon name.

**redlistCategory** Red List threat category

**authority** Taxon author/s.

**Source**

Downloaded and sampled from <https://www.iucnredlist.org/>

---

taxonomic\_mapping      *Data for mapping plant family to order or higher classification*

---

**Description**

A dataset containing the higher classification (Angiosperms, Gymnosperms, Ferns and Lycophytes) and Order for each family in the WCVP.

**Usage**

taxonomic\_mapping

**Format**

A data frame with 457 rows and 3 variables: family, order and higher\_classification

**Source**

Fern and lycophyte taxonomy from PPG I (2016; [doi:10.1111/jse.12229](https://doi.org/10.1111/jse.12229)). Angiosperm taxonomy from APG IV (2016; [doi:10.1111/boj.12385](https://doi.org/10.1111/boj.12385)). Gymnosperm taxonomy from Forest et al (2018; [doi:10.1038/s41598018243654](https://doi.org/10.1038/s41598018243654))

---

wcvp_checklist	<i>Generate a species checklist from WCVF</i>
----------------	---

---

### Description

Generate a species checklist from WCVF

### Usage

```
wcvp_checklist(
  taxon = NULL,
  taxon_rank = c("species", "genus", "family", "order", "higher"),
  area_codes = NULL,
  synonyms = TRUE,
  render_report = FALSE,
  native = TRUE,
  introduced = TRUE,
  extinct = TRUE,
  location_doubtful = TRUE,
  hybrids = FALSE,
  infraspecies = TRUE,
  report_filename = NULL,
  report_dir = NULL,
  report_type = c("alphabetical", "taxonomic"),
  wcvp_names = NULL,
  wcvp_distributions = NULL
)
```

### Arguments

taxon	Character. Taxon to be included. Defaults to NULL (no taxonomic filter; all taxa).
taxon_rank	Character. One of "species", "genus", "family", "order" or "higher", giving the rank of the value/s in taxon. Must be specified unless taxon is NULL.
area_codes	Character. One or many WGSPRD level 3 region codes. Defaults to NULL (global).
synonyms	Logical. Include synonyms in checklist (see Details)? Defaults to TRUE.
render_report	Logical. Render the checklist as a markdown report? Defaults to FALSE.
native	Logical. Include species occurrences not flagged as introduced, extinct or doubtful? Defaults to TRUE.
introduced	Logical. Include species occurrences flagged as introduced? Defaults to TRUE.
extinct	Logical. Include species occurrences flagged as extinct? Defaults to TRUE.
location_doubtful	Logical. Include species occurrences flagged as location_doubtful? Defaults to TRUE.

hybrids	Logical. Include hybrid species in checklist? Defaults to FALSE.
infraspecies	Logical. Include hybrid species in checklist? Defaults to TRUE.
report_filename	Character. Name for the HTML file. Defaults to taxon_area_type.html
report_dir	Character. Directory for the HTML file to be saved in. Must be provided by user.
report_type	Character; one of "alphabetical" (the default) or "taxonomic". Should the generated checklist be sorted alphabetically, or by taxonomic status?
wcvp_names	A data frame of taxonomic names from WCVP version 7 or later. If NULL (the default), names will be loaded from <code>rWCVPdata::wcvp_names</code> .
wcvp_distributions	A data frame of distributions from WCVP version 7 or later. If NULL (the default), distributions will be loaded from <code>rWCVPdata::wcvp_names</code> .

### Details

The `synonyms` argument can be used to limit names to those that are Accepted. If `synonyms = TRUE` then invalid, illegitimate and other non-accepted names are also included (i.e., the checklist is not limited to names for which `taxon_status == "Synonym"`). Two styles of checklist are supported in `rWCVP` - alphabetical and taxonomic. In an alphabetical checklist, all names are arranged alphabetically with accepted names in bold, and synonyms are followed by their accepted name. For a taxonomic checklist, names are grouped by their accepted names, and synonyms are listed beneath. Both types of checklist include author, publication and distribution information, though note that family headings are only supported in alphabetical checklists (due to the additional grouping requirement of the taxonomic format).

### Value

Data frame with filtered data and, if `render_report=TRUE`, a report HTML file.

### Examples

```
# These examples take >10 seconds to run and require 'rWCVPdata'

if(requireNamespace("rWCVPdata")){
  wcvp_checklist(taxon = "Myrtaceae", taxon_rank = "family", area = get_wgsrpd3_codes("Brazil"))
  wcvp_checklist(taxon = "Ferns", taxon_rank = "higher", area = get_wgsrpd3_codes("New Zealand"))
}
```

---

wcvp\_distribution      *Generate spatial distribution objects for species, genera or families*

---

### Description

Generate spatial distribution objects for species, genera or families



**Usage**

```
wcvp_distribution(
  taxon,
  taxon_rank = c("species", "genus", "family", "order", "higher"),
  native = TRUE,
  introduced = TRUE,
  extinct = TRUE,
  location_doubtful = TRUE,
  wcvp_names = NULL,
  wcvp_distributions = NULL
)
```

**Arguments**

taxon	Character. The taxon to be mapped. Must be provided.
taxon_rank	Character. One of "species", "genus", "family", "order" or "higher", giving the rank of the value in taxon.
native	Logical. Include native range? Defaults to TRUE.
introduced	Logical. Include introduced range? Defaults to TRUE.
extinct	Logical. Include extinct range? Defaults to TRUE.
location_doubtful	Logical. Include occurrences that are thought to be doubtful? Defaults to TRUE.
wcvp_names	A data frame of taxonomic names from WCVP version 7 or later. If NULL (the default), names will be loaded from <code>rWCVPdata::wcvp_names</code> .
wcvp_distributions	A data frame of distributions from WCVP version 7 or later. If NULL (the default), distributions will be loaded from <code>rWCVPdata::wcvp_names</code> .

**Details**

Where `taxon_rank` is higher than species, the distribution of the whole group will be returned, not individual species within that group. This also applies when toggling options - for example, introduced occurrences will only be included if they are outside the native range, regardless of whether `native=TRUE` or `native=FALSE`. To identify extinctions, introductions or doubtful occurrences within the native range, the `wcvp_summary` and `wcvp_occ_mat` functions can be used.

**Value**

Simple features (sf) data frame containing the range polygon/s of the taxon.

**Examples**

```
# this example requires 'rWCVPdata'
if(requireNamespace("rWCVPdata")){
  r <- wcvp_distribution("Callitris", taxon_rank = "genus")
  p <- wcvp_distribution_map(r)
  p
}
```

```
}
```

---

wcvp\_distribution\_map *Plot distribution maps for species, genera or families*

---

## Description

Plot distribution maps for species, genera or families

## Usage

```
wcvp_distribution_map(  
  range,  
  crop_map = FALSE,  
  native = TRUE,  
  introduced = TRUE,  
  extinct = TRUE,  
  location_doubtful = TRUE  
)
```

## Arguments

range	Simple features (sf) data frame of the type output by <code>wcvp_distribution()</code> .
crop_map	Logical. Crop map extent to distribution? Defaults to FALSE.
native	Logical. Include native range? Defaults to TRUE.
introduced	Logical. Include introduced range? Defaults to TRUE.
extinct	Logical. Include extinct range? Defaults to TRUE.
location_doubtful	Logical. Include occurrences that are thought to be doubtful? Defaults to TRUE.

## Details

The colour scheme mirrors that used by Plants of the World (POWO; <https://powo.science.kew.org/>), where green is native, purple is introduced, red is extinct and orange is doubtful. See Examples for how to use custom colours.

## Value

A `ggplot2::ggplot` of the distribution.

**Examples**

```

# these examples require 'rWCVpdata'
if(requireNamespace("rWCVpdata")){
p <- wcvp_distribution_map(wcvp_distribution("Callitris", taxon_rank = "genus"))
p
# now only the native range, and cropped to range extent
p <- wcvp_distribution_map(wcvp_distribution("Callitris", taxon_rank = "genus"),
  introduced = FALSE, crop_map = TRUE
)
p
# now with different colours
# note that this taxon only has native and introduced occurrences, so only two colours are needed
p <- wcvp_distribution_map(wcvp_distribution("Callitris", taxon_rank = "genus"))
p +
  # for polygons
  ggplot2::scale_fill_manual(values = c("red", "blue")) +
  # for points (islands)
  ggplot2::scale_colour_manual(values = c("red", "blue"))
}

```

---

wcvp_match_exact	<i>Exact matching to WCVP.</i>
------------------	--------------------------------

---

**Description**

Exact matching of names to the WCVP, optionally using the author string to refine results.

**Usage**

```
wcvp_match_exact(names_df, wcvp_names, name_col, author_col = NULL, id_col)
```

**Arguments**

names_df	Data frame of names for matching.
wcvp_names	Data frame of taxonomic names from WCVP version 7 or later. If NULL (the default), names will be loaded from <code>rWCVpdata::wcvp_names</code> .
name_col	Character. The column in names_df that has the taxon name for matching.
author_col	the column in names_df that has the name authority, to aid matching. Set to NULL to match with no author string.
id_col	the column in names_df that has the observation id.

**Value**

Match results from WCVP bound to the original data from names\_df.

**See Also**

Other name matching functions: [wcvp\\_match\\_fuzzy\(\)](#), [wcvp\\_match\\_names\(\)](#)

**Examples**

```
# these examples require 'rWCVpdata'
if(requireNamespace("rWCVpdata")){
wcvp_names <- rWCVpdata::wcvp_names

# including author string
wcvp_match_exact(redlist_example, wcvp_names, "scientificName",
  author_col = "authority",
  id_col = "assessmentId"
)

# without author string
wcvp_match_exact(redlist_example, wcvp_names, "scientificName", id_col = "assessmentId")
}
```

---

wcvp\_match\_fuzzy

*Fuzzy (approximate) matching to the WCVP.*


---

**Description**

Fuzzy matching to names in the WCVP using phonetic matching and edit distance. The WCVP can be loaded for matching from [rWCVpdata::wcvp\\_names](#).

**Usage**

```
wcvp_match_fuzzy(names_df, wcvp_names, name_col, progress_bar = TRUE)
```

```
phonetic_match(names_df, wcvp_names, name_col)
```

```
edit_match(names_df, wcvp_names, name_col)
```

**Arguments**

names_df	Data frame of names for matching.
wcvp_names	Data frame of taxonomic names from WCVP version 7 or later. If NULL (the default), names will be loaded from <a href="#">rWCVpdata::wcvp_names</a> .
name_col	Character. The column in names_df that has the taxon name for matching.
progress_bar	Logical. Show progress bar when matching? Defaults to TRUE; should be changed to FALSE if used in a markdown report.

## Details

The `wcvp_match_fuzzy` function uses phonetic matching first and then finds the closest match based on edit distance for any remaining names.

Phonetic matching uses [phonics::metaphone](#) encoding with a maximum code length of 20.

Edit distance matching finds the closest match based on Levenshtein similarity, calculated by [RecordLinkage::levenshteinSim](#).

## Value

Match results from WCVP bound to the original data from `names_df`.

## See Also

Other name matching functions: [wcvp\\_match\\_exact\(\)](#), [wcvp\\_match\\_names\(\)](#)

## Examples

```
# this example requires 'rWCVpdata'
if(requireNamespace("rWCVpdata")){
  wcvp_names <- rWCVpdata::wcvp_names
  wcvp_match_fuzzy(redlist_example, wcvp_names, "scientificName")
}
```

```
# this example requires 'rWCVpdata'
if(requireNamespace("rWCVpdata")){
  wcvp_names <- rWCVpdata::wcvp_names
  phonetic_match(redlist_example, wcvp_names, "scientificName")
}
```

```
# this example requires 'rWCVpdata'
if(requireNamespace("rWCVpdata")){
  wcvp_names <- rWCVpdata::wcvp_names
  edit_match(redlist_example, wcvp_names, "scientificName")
}
```

---

`wcvp_match_names`

*Match names to the WCVP.*

---

## Description

Match names to WCVP, first using exact matching and then using fuzzy matching on any remaining unmatched names.

**Usage**

```
wcvp_match_names(
  names_df,
  wcvp_names = NULL,
  name_col = NULL,
  id_col = NULL,
  author_col = NULL,
  join_cols = NULL,
  fuzzy = TRUE,
  progress_bar = TRUE
)
```

**Arguments**

<code>names_df</code>	Data frame of names for matching.
<code>wcvp_names</code>	Data frame of taxonomic names from WCVP version 7 or later. If NULL (the default), names will be loaded from <code>rWCVpdata::wcvp_names</code> .
<code>name_col</code>	Character. The column in <code>names_df</code> that has the taxon name for matching.
<code>id_col</code>	Character. A column in <code>names_df</code> with a unique ID for each name. Will be created from the row number if not provided.
<code>author_col</code>	the column in <code>names_df</code> that has the name authority, to aid matching. Set to NULL to match with no author string.
<code>join_cols</code>	Character. A vector of name parts to make the taxon name, if <code>name_col</code> is not provided.
<code>fuzzy</code>	Logical; whether or not fuzzy matching should be used for names that could not be matched exactly.
<code>progress_bar</code>	Logical. Show progress bar when matching? Defaults to TRUE; should be changed to FALSE if used in a markdown report.

**Details**

By default, exact matching uses only the taxon name (supplied by `name_col`) unless a column specifying the author string is provided (as `author_col`).

Columns setting out name parts can be supplied as `join_cols` in place of a taxon name, but must be supplied in the order you want them joined (e.g. `c("genus", "species", "infra_rank", "infra")`).

Fuzzy matching uses a combination of phonetic and edit distance matching, and can optionally be turned off using `fuzzy=FALSE`.

The WCVP can be loaded for matching from `rWCVpdata::wcvp_names`.

See [here](#) for an example workflow.

**Value**

Match results from WCVP bound to the original data from `names_df`.

**See Also**

Other name matching functions: [wcvp\\_match\\_exact\(\)](#), [wcvp\\_match\\_fuzzy\(\)](#)

**Examples**

```
# these examples require 'rWCVpdata'
if(requireNamespace("rWCVpdata")){
  wcvp_names <- rWCVpdata::wcvp_names

  # without author
  wcvp_match_names(redlist_example, wcvp_names,
    name_col = "scientificName",
    id_col = "assessmentId"
  )

  # with author
  wcvp_match_names(redlist_example, wcvp_names,
    name_col = "scientificName",
    id_col = "assessmentId", author_col = "authority"
  )
}
```

---

wcvp\_occ\_mat

*Generate occurrence matrix for taxa and areas*

---

**Description**

Generate occurrence matrix for taxa and areas

**Usage**

```
wcvp_occ_mat(
  taxon = NULL,
  taxon_rank = c("species", "genus", "family", "order", "higher"),
  area_codes = NULL,
  native = TRUE,
  introduced = TRUE,
  extinct = TRUE,
  location_doubtful = TRUE,
  wcvp_names = NULL,
  wcvp_distributions = NULL
)
```

**Arguments**

taxon	Character. One or many taxa to be included. Defaults to NULL (all species)
taxon_rank	Character. One of "species", "genus", "family", "order" or "higher", giving the rank of the value/s in taxon. Must be specified unless taxon is NULL.
area_codes	Character. One or many WGSPRD level 3 region codes. Defaults to NULL (global).
native	Logical. Include species occurrences not flagged as introduced, extinct or doubtful? Defaults to TRUE.
introduced	Logical. Include species occurrences flagged as introduced? Defaults to TRUE.
extinct	Logical. Include species occurrences flagged as extinct? Defaults to TRUE.
location_doubtful	Logical. Include species occurrences flagged as location doubtful? Defaults to TRUE.
wcvp_names	A data frame of taxonomic names from WCVP version 7 or later. If NULL, names will be loaded from <code>rWCVPdata::wcvp_names</code> .
wcvp_distributions	A data frame of distributions from WCVP version 7 or later. If NULL, distributions will be loaded from <code>rWCVPdata::wcvp_names</code> .

**Details**

See [here](#) for an example of how this output can be formatted for publication.

**Value**

A data.frame containing the `taxon_name` and `plant_name_id` of all species that are present in the area, plus one variable for each WGSPRD level 3 region in area, with species presences marked as 1 and absences marked as 0.

**Examples**

```
# this example requires 'rWCVPdata'
if(requireNamespace("rWCVPdata")){
wcvp_occ_mat(
  taxon = "Poa", taxon_rank = "genus",
  area = c("TAS", "VIC", "NSW"), introduced = FALSE
)
}
```



---

wcvp_reformat	<i>Reformat local versions of WCVP</i>
---------------	--

---

**Description**

Reformat local versions of WCVP

**Usage**

```
wcvp_reformat(wcvp_local, version = NULL)
```

**Arguments**

wcvp_local	Data.frame. Local copy of the WCVP.
version	Either 9 or "v9". We will add support for other versions as needed.

**Details**

Note that not all of the original variables are preserved during reformatting. For example, publication is a single variable in v9, but split over multiple in the data package. It is therefore not possible to simply rename this variable. Variables that are present in the data package but not in v9 are filled with NA.

**Value**

A data.frame with the same variable structure as the WCVP that is included in the data package rWCVpdata.

---

wcvp_summary	<i>Generate a summary table from the WCVP</i>
--------------	---

---

**Description**

Generate a summary table from the WCVP

**Usage**

```
wcvp_summary(
  taxon = NULL,
  taxon_rank = c("species", "genus", "family", "order", "higher"),
  area_codes = NULL,
  grouping_var = c("area_code_l3", "genus", "family", "order", "higher"),
  hybrids = FALSE,
  wcvp_names = NULL,
  wcvp_distributions = NULL
)
```

**Arguments**

taxon	Character. Taxon to be included. Defaults to NULL (no taxonomic filter; all taxa).
taxon_rank	Character. One of "genus", "family", "order" or "higher", giving the rank of the value/s in taxon. Must be specified unless taxon is NULL.
area_codes	Character. One or many WGSRPD level 3 region codes. Defaults to NULL (global).
grouping_var	Character; one of "area_code_l3", "genus", "family", "order" or "higher" specifying how the summary should be arranged. Defaults to area_code_l3.
hybrids	Logical. Include hybrid species in counts? Defaults to FALSE.
wcvp_names	A data frame of taxonomic names from WCVF version 7 or later. If NULL, names will be loaded from <code>rWCVFdata::wcvp_names</code> .
wcvp_distributions	A data frame of distributions from WCVF version 7 or later. If NULL, distributions will be loaded from <code>rWCVFdata::wcvp_names</code> .

**Details**

Valid values for rank 'higher' are 'Angiosperms', 'Gymnosperms', 'Ferns' and 'Lycophytes'. Note that grouping variable (if taxonomic) should be of a lower level than taxon and taxon\_rank to produce a meaningful summary (i.e., it does not make sense to group a genus by genus, family or higher classification). Additionally, if the grouping variable is taxonomic then species occurrences are aggregated across the input area. This means that if a species is native to any of the input area (even if it is introduced or extinct in other parts) it is counted as 'Native'. Similarly, introduced occurrences take precedence over extinct occurrences. Note that in this type of summary table, 'Endemic' means endemic to the input area, not necessarily to a single WGSRPD Level 3 Area within the input area.

**Value**

Data.frame with filtered data, or a gt table

**Examples**

```
# this example requires 'rWCVFdata'
if(requireNamespace("rWCVFdata")){
  ferns <- wcvp_summary("Ferns", "higher", get_wgsrpd3_codes("New Zealand"), grouping_var = "family")
  wcvp_summary_gt(ferns)
}
```

---

wcvp_summary_gt	<i>Render a summary table from <a href="#">wcvp_summary</a></i>
-----------------	---

---

### Description

Render a summary table from [wcvp\\_summary](#)

### Usage

```
wcvp_summary_gt(x)
```

### Arguments

x                   List.

### Value

[gt](#) table

### Examples

```
# this example requires 'rWCVPdata'  
if(requireNamespace("rWCVPdata")){  
  ferns <- wcvp_summary("Ferns", "higher", get_wgsrpd3_codes("New Zealand"), grouping_var = "family")  
  wcvp_summary_gt(ferns)  
}
```

---

wgsrpd3	<i>Biodiversity Information Standards (TDWG) World Geographical Scheme for Recording Plant Distributions (WGSRPD)</i>
---------	---

---

### Description

Spatial data for WGSRPD Level 3, for plotting maps

### Usage

```
wgsrpd3
```

**Format**

An 'sf' object with 20 rows and 4 variables:

**LEVEL3\_NAM** Region name  
**LEVEL3\_COD** Region code  
**LEVEL2\_COD** Level 2 code  
**LEVEL1\_COD** Level 1 code (continent)  
**geometry** sf geometry  
**fillcol** Used for mapping.

**Source**

<https://github.com/tdwg/wgsrpd/tree/master/level3>

---

wgsrpd\_mapping

*Data for mapping WGSRPD geography to other levels*

---

**Description**

A dataset containing the area (Level 3), #' region (Level 2), continent (Level 1), country (political) and hemisphere category for each Level 3 area. Country mapping follows Gallagher et al. (2020).

**Usage**

wgsrpd\_mapping

**Format**

A data frame with 370 rows and 7 variables:

**HEMISPHERE** Northern, Southern or Equatorial (spanning the equator).  
**LEVEL1\_COD** Continent code.  
**LEVEL1\_NAM** Continent.  
**LEVEL2\_COD** Region code.  
**LEVEL2\_NAM** Region.  
**COUNTRY** Country (political; from Gallagher et al., 2020)  
**LEVEL3\_COD** Area code.  
**LEVEL3\_NAM** Area.

**Source**

Modified from data available at <https://github.com/tdwg/wgsrpd>

# Index

- \* **datasets**
  - redlist\_example, [5](#)
  - taxonomic\_mapping, [6](#)
  - wgsrpd3, [19](#)
  - wgsrpd\_mapping, [20](#)
- \* **name matching functions**
  - wcvp\_match\_exact, [11](#)
  - wcvp\_match\_fuzzy, [12](#)
  - wcvp\_match\_names, [13](#)
- wcvp\_match\_exact, [11](#), [13](#), [15](#)
- wcvp\_match\_fuzzy, [12](#), [12](#), [15](#)
- wcvp\_match\_names, [12](#), [13](#), [13](#)
- wcvp\_occ\_mat, [15](#)
- wcvp\_reformat, [17](#)
- wcvp\_summary, [17](#), [19](#)
- wcvp\_summary\_gt, [19](#)
- wgsrpd3, [19](#)
- wgsrpd\_mapping, [20](#)

discrete\_scale, [4](#)

edit\_match (wcvp\_match\_fuzzy), [12](#)

get\_area\_name, [2](#)

get\_wgsrpd3\_codes, [3](#)

ggplot2::ggplot, [10](#)

gt, [19](#)

guides(), [5](#)

lambda, [5](#)

phonetic\_match (wcvp\_match\_fuzzy), [12](#)

phonics::metaphone, [13](#)

powo\_map, [4](#)

powo\_pal, [4](#)

RecordLinkage::levenshteinSim, [13](#)

redlist\_example, [5](#)

scale\_color\_powo (powo\_pal), [4](#)

scale\_colour\_powo (powo\_pal), [4](#)

scale\_fill\_powo (powo\_pal), [4](#)

scales::hue\_pal(), [4](#)

taxonomic\_mapping, [6](#)

wcvp\_checklist, [7](#)

wcvp\_distribution, [8](#)

wcvp\_distribution(), [10](#)

wcvp\_distribution\_map, [10](#)